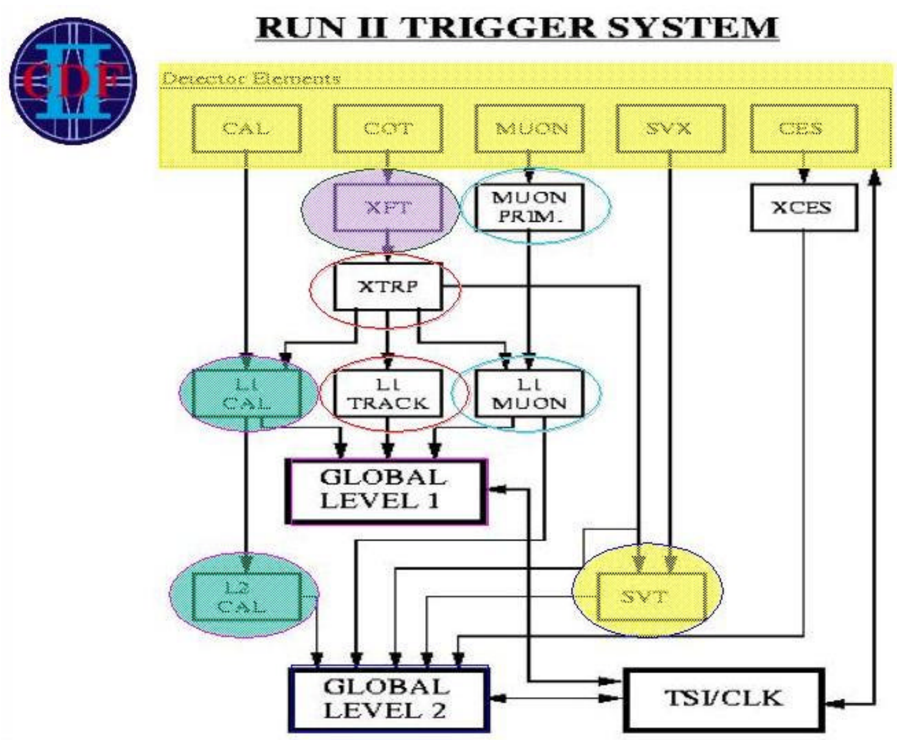
TRGSim++ status

TRGSim++ is a set of (C++) packages emulating the various trigger levels decision steps (CDF trigger is fully digital)



Trigger decision steps: A_C++ modules, organized in packages:

CalTrigger -> CalTriggerExe
MuonTrigger -> MuonTriggerExe
XFTSim ->XFTTest (tbin)
svtim -> svtsimtest (tbin)
XTRPSim -> XTRPSimExe(tbin)
L2GlobalTrigger ->L2Sim
L1GlobalTrigger ->FredSim
TriggerMods -> TRGSim++
TriggerObjects -> trigger banks

TRGSim++ modules run off detector raw data and produce emulated trigger data identical to real hardware data.

It runs off COTQ, CalData also

Documentation and examples

http://ncdf70.fnal.gov:8001/trigsim/trgsim.html

General Information;
Information on the code : what it does and how to run (link to most recent tcl's).

To run TRGSim++ ALWAYS go and look at the following tcl:

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++.tcl

This is the only tcl I'm constantly updating for users reference

Status: CalTrigger

- Trigger Tower energies (TC2D)
- L1 DIRAC Triggers (TC1D)
- L2 clustering and Iso sums (TC2D)
- database access for trigger definition - real data
  - TriggerDB DOWNLOADS Table
  - possibility to run on simulated run with conditions from real run

CalTrigger input/output

CalTrigger reads in input the calorimeter D-banks: CEMD, CHAD, PEMD, PHAD, WHAD. It alternatively reads CalData and puffs it into D-banks, via a call to CalorimetryModule.

It also gets the track input from XTRD_StorableBank.

There are 2 options to get the track input controlled by the parameter "standalone" in the talk-to: from real data (standalone = 1) or from a simulated XTRD (standalone = 0).

CalTrigger produces in output the following banks:

- TC2D with the list of trigger tower energies and clustering summary word for each tower (description "Simulated Trigger Bank".
- TC1D with the results of the DIRAC triggers: description "Simulated Trigger Bank".
- TL1D with the PreFred bits for CalTrigger and SumEt: description "Simulated Trigger Bank".
- TL2D where the only block filled is the clustering block: description "Calorimeter TL2D Bank".

Status XFTSim

- XFT tracks :
  - XFLD and XFFD diagnostic banks
  - XTRD with track parameter
- DB access implemented
  - real run conditions applied to simulated run:
- For the Road and Mask files one can set the file types. So you can mimic exactly what was run. This is not automated so one has to do it by hand. The other aspect is the deadwires. Here the answer is definitely "No". Try to add this feature in the near future.

XFT input/output

- Input:
  - COTD hit information (void XFTSim::getInputData())
  - COTQ if COTD is not found in data stream
- Output:
  - XFFD - XFT Finder bank
  - XFLD - XFT Linker bank

Status : XTRPSim

- XTRP tracks:
  - XTRD
- db access implementation ongoing
  - thresholds are hardwired for the moment

XTRPSim

- Input : XFLD (real or simulated controlled via talk-to parameter “ *standalone*”)
- output XTRD bank:
  - track data: 2 blocks:
    - first block: 12 cards (2 wedges per card)
      - Track data:
        - ϕ
        - p_T
        - Isolation bit
        - short track bit
        - XTRP/Calorimeter bus
    - 2nd block: track trigger bits word

Status: MuonTrigger

- Muon primitives:
  - TCMD
  - Xtrapolation with XTRP done
  - database access not implemented
    - stubs thresholds hardwired

MuonTrigger input/output

- Input:
  - CMUD
  - CMXD
  - CMPD
  - XTRPMatchBoxData
- Output:
  - TCMD

Status: svtsim

- SVT tracks:
  - SVTD
- database access implemented
- Input SIXD and XFLD

FredSim

- L1GlobalTrigger/src/L1GlobalTriggerMaker.cc
  - reads in FRED bank, 1st and 2nd words (input to FRED)
  - reads in TL1D bank, PreFred words
  - constructs simulated bits from pattern in TriggerDB (class DBFredAccess)
  - simulate unprescaled bits and prescaled bits (3-4 and 5-6 words)
  - output simulated TFRD bank

Status: L2GlobalTrigger

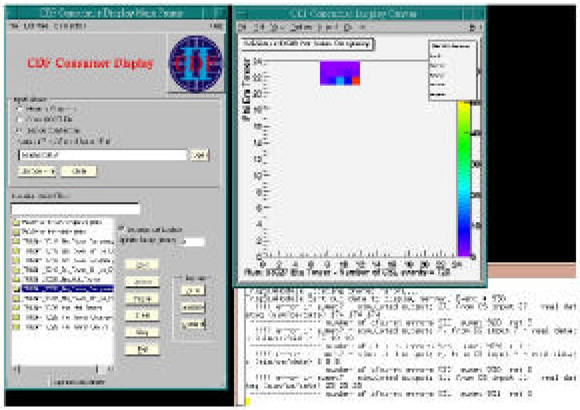
- L2Sim: L2 bits + L2 quantities
  - TL2D simulated fills only:
    - cluster block
    - SVT block
  - no db access

Monitoring the trigger online

- Each subsystem driver module contains data monitors methods which check word by word real and simulated banks, firing error messages on the error logger and on the standard output if discrepancy is found:
  - TC1D/TC2D_monitor (AbsEvent* anEvent)
  - XFFD/XFLD_monitor(AbsEvent* anEvent)
  - XTRD_monitor(AbsEvent* anEvent) SVTD_monitor(AbsEvent* anEvent)
  - TCMD_monitor(AbsEvent* anEvent)
- Summary files produced for every run, control on level of printing
- Easily coupled to the online ROOT-based Consumer Framework (HistoDisplayMain)

in TRIGMON and standalone

Online monitoring



TRGSim++/CalTrigger is a powerful monitoring tool to spot:

Hot Cables: no need to mask a tower when there is a loose cable between boards
BC mismatches found immediately. (also bits stuck low)
Admem ET weighting errors
Swapped cables are also found immediately
Unmasking Hot Towers

Trigger banks

- TriggerObjects
  - trigger banks as in 4152
  - simple accessors (get_word (int, int, int)
  - named accessors (depending on the bank)
    - look at the code browser
  - examples: TriggerObjects/src/Writer.cc, TriggerObjects/src/Reader.cc
- Most of the trigger banks are in evtNtuple:
  - trigger towers energies, MET and SumEt
  - XTRP tracks
  - SVT track
  - L1 trigger bits (prescaled and uprescaled)

  http://ncdf70.fnal.gov:8001/talks/eN/eN.html

Summary

- TRGSim emulates the hardware trigger:
  - Calorimeter trigger :
    - TC2D trigger towers energies
    - TC1D DIRAC trigger words (ele, gamma,jet,tau)
    - TL1D Sumet/MET
    - TL2D cluster block
  - XFTSim/XTRPSim
    - XTRD track list
  - MuonTrigger
    - TCMD
  - svtsim
    - SVTD
  - FredSim L2Sim
    - TFRD TL2D